Chapter = 8 =

Discrete Fourier Transform (DFT).

* We discussed representation of sequences and LTI systems in terms of Fourier transform and Z-transform.
  We can convert convolution in time-domain to multiplication in freq. domain and Z-domain.

* Implementing DT systems by computing Fourier transform of two sequences and multiplying them together and then find inverse Fourier transform is hard to imagine, because FT is continuous function of ω. Which means, we have to compute infinite number of frequencies.

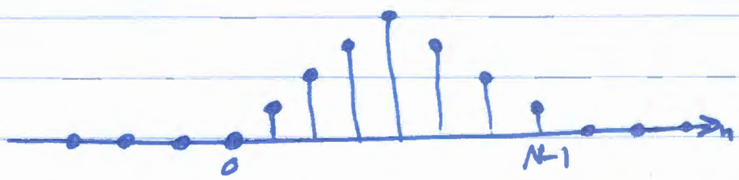
* In this chapter, we introduce DFT for a finite-duration sequence.

* DFT is related to Discrete-time Fourier transform $\Rightarrow$ DFT is samples of DT Fourier transform or more generally samples of Z-transform. That requires to impose restrictions on the samples.

* DFT has some properities same as DT-FT and some are different.
* We relate DFT to Discrete Fourier Series (DFS).

$x(n)$ = finite-length (N)

$\Rightarrow$ we can construct periodic seq. $\tilde{x}(n)$



$$\tilde{x}(n) = x(n) + x(n+N) + \dots = \sum_{r=-\infty}^{\infty} x(n+rN)$$

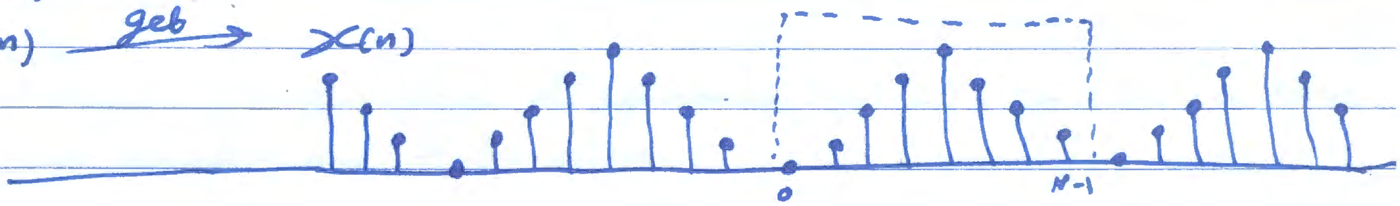
$$x(n \text{ modulo } N)$$

$$\triangleq x((n))_N$$

$x(n) \xrightarrow{get} \tilde{x}(n)$

$\tilde{x}(n) \xrightarrow{get} x(n)$

$R_N(n)$



- Finite-length $x(n)$ → specifies by N samples.

- $\tilde{x}(n)$ → specifies also by N samples ~~[illegible]~~. (repeating them again and again).

* Display finite-samples around a cylinder instead of linear.

$$x(n) = \begin{cases} \tilde{x}(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

$$x(n) = \tilde{x}(n) R_N(n)$$

$$R_N(n) = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

Why?

In CT periodic signals → represented by Fourier Series, and periodic DT signals can be also represented by Fourier Series.

$\tilde{x}(n)$ has a Fourier Series representation

DFS of $\tilde{x}(n)$ $\triangleq$ DFT of $x(n)$.

Discrete Fourier Series (DFS)

$\tilde{x}(n)$: periodic seq. with period N.

$$\tilde{x}(n) = \sum \tilde{X}(k)\, e^{j\frac{2\pi}{N}nk} \rightarrow$$ linear Combination of harmonically related complex exp.

same as in Continous-time domain.

* how many distinct harmonically complex exponentials

$$e^{j\frac{2\pi}{N}nk} = \underbrace{e^{j\frac{2\pi}{N}n(k+N)}} = e^{j\frac{2\pi}{N}nk} \cdot \underbrace{e^{j\frac{2\pi}{N}nN}}_{=1}.$$

we know periodic in (n).

$\Rightarrow$ also periodic in (k).

* In DT domain, as we vary sinusoids in frequency, we see same values again in Interval $0 \rightarrow 2\pi$.

* It means, in forming DFS, once we use complex exponentials for k between 0 and $N-1$, we use all complex exponentials with this fundamental freq. we have. and if we keep going with k, we just see same complex exponentials over and over again.

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k)\, e^{+j\frac{2\pi}{N}nk}$$ (Called analysis equation)

normalization factor $\leftarrow \frac{1}{N}$
(plays same role of $\frac{1}{2\pi}$ in DT.FT

these coeff. are periodic $\leftarrow$ $$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n)\, e^{-j\frac{2\pi}{N}nk}$$ (synthesis eq.)

* We said, there are finite number of complex exponentials (distinct coefficients) between 0 and $(N-1)$, we see all ones

$\Rightarrow$ so, there are finite distinct Fourier Series Coefficients.

Fact: $e^{-j\frac{2\pi}{N}n(k+N)} = e^{-j\frac{2\pi}{N}nk}$

$$\tilde{X}(k+N) = \tilde{X}(k) = \tilde{X}(k+2N) + \ldots$$

so, it is periodic seq. $\Rightarrow$ so, we use coeff. of one period.

$\tilde{X}(k)$ periodic in k of period N
$\tilde{x}(n)$ " " n " " " .

* $\Rightarrow$ There is duality between time-domain and freq. domain

$$\boxed{W_N \triangleq e^{-j\frac{2\pi}{N}}}$$

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) W_N^{-nk}$$

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) W_N^{nk}$$

DFS properities:

* Shifting

$$\tilde{x}(n+m) \xleftrightarrow{DFS} W_N^{-km} \tilde{X}(k)$$

$$W_N^{Ln} \tilde{x}(n) \xleftrightarrow{DFS} \tilde{X}(k+L)$$

* symmetry: $\tilde{x}(n)$ real

$$\tilde{X}(k) = \tilde{X}_R(k) + j\,\tilde{X}_I(k)$$

$\tilde{X}_R(k) = X_R(-k)$ (even)
$X_R(k) = \tilde{X}_R(N-k)$ periodic

$$\tilde{X}_I(k) = -\tilde{X}_I(k) \quad (\text{odd})$$
$$= -\tilde{X}(N-k)$$

$|\tilde{X}(k)| \rightarrow$ even (function of k).

$\angle \tilde{X}(k) \rightarrow$ odd function of k.

* Convolution property

$$\tilde{x}_1(n) \longleftrightarrow \tilde{X}_1(k)$$
$$\tilde{x}_2(n) \longleftrightarrow \tilde{X}_2(k)$$
$$\tilde{x}_3(n) \longleftrightarrow \tilde{X}_3(k) = \tilde{X}_1(k)\,\tilde{X}_2(k)$$

What is periodic seq. $\tilde{x}_3(n)$ whose DFS is $\tilde{X}_3(k)$?

$$\tilde{x}_3(n) = \sum_{m=0}^{N-1} \tilde{x}_1(m)\,\tilde{x}_2(n-m) \quad (\text{periodic Convolution})$$

* One difference from DT-Fourier transform and Z-transform that sum from 0 to $N-1$ and not $-\infty \rightarrow +\infty$. (only over one period).

* Dual property

$$\tilde{x}_4(n) = \tilde{x}_1(n)\,\tilde{x}_2(n)$$
$$\tilde{X}_4(k) = \frac{1}{N}\sum_{L=0}^{N-1} \tilde{X}_1(L)\,\tilde{X}_2(k-L)$$

* periodic Convolution

Example: (impulse Train)

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} \delta(n-rN) = \begin{cases} 1, & n = rN, \quad r \rightarrow \text{integer.} \\ 0, & \text{otherwise} \end{cases}$$

since $\tilde{x}(n) = \delta(n)$ for $0 \le n \le N-1 \Rightarrow$ DFS coeff. $\tilde{X}(k)$

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \delta(n) W_N^{kn} = W_N^0 = 1 \text{ for all } k.$$

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} \delta(n-rN) = \frac{1}{N}\sum_{k=0}^{N-1} W_N^{-kn} = \frac{1}{N}\sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}kn}$$

* so, periodic Impulse train $\rightarrow$ represented in terms of sum of complex exponentials, where, all complex exp. have the same magnitude and phase, and ~~add~~ all add to unity at Integer multiple of N and to zero for all other Integers.

Example: Duality

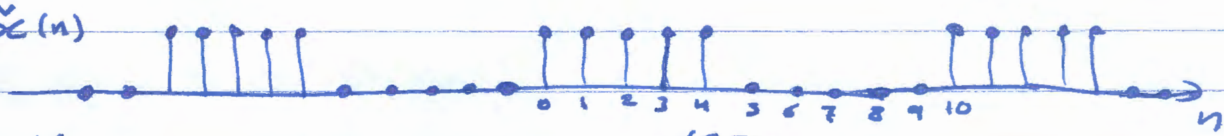
$$\tilde{Y}(k) = \sum_{r=-\infty}^{\infty} N\delta(k-rN) \quad \text{DFS coeff. Impulse train}$$

$$\tilde{y}(n) = \frac{1}{N}\sum_{k=0}^{N-1} N\delta(k) W_N^{-kn} = W_N^{-0} = 1 \quad \text{for all } n.$$

comparing this result with result of previous example $\Rightarrow$

$$\tilde{Y}(k) = N\tilde{x}(k) \text{ and } \tilde{y}(n) = \tilde{X}(n).$$

Example: $\tilde{x}(n)$



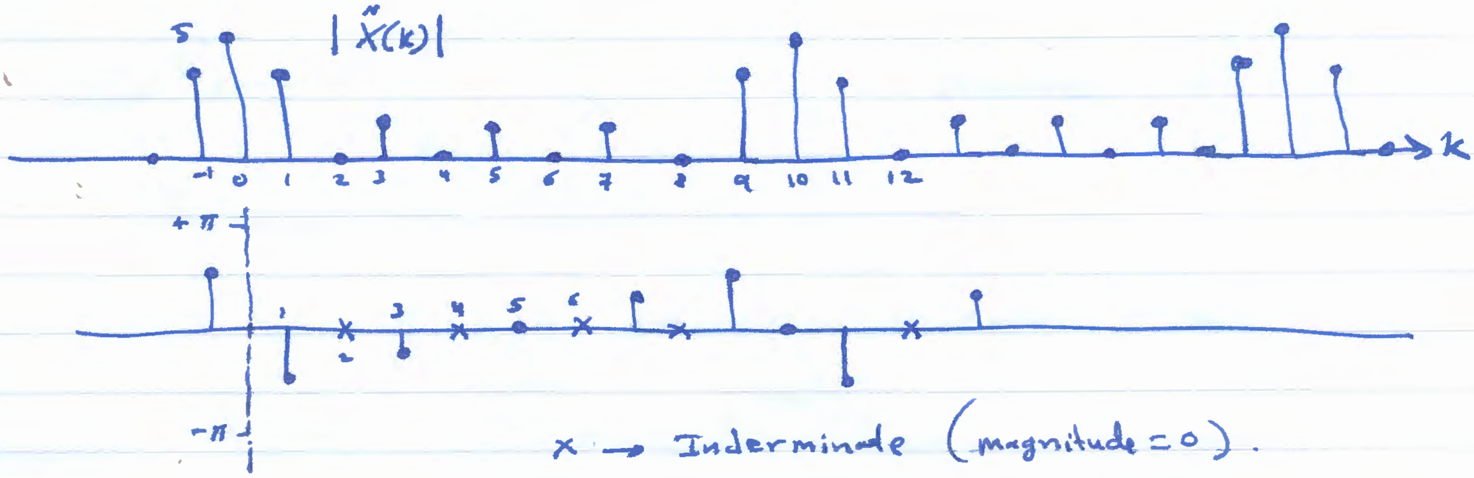
$N = 10$

$$\tilde{X}(k) = \sum_{n=0}^{4} W_{10}^{kn} = \sum_{n=0}^{4} e^{-j\left(\frac{2\pi}{10}\right)kn}$$

$$= \frac{1 - W_{10}^{5k}}{1 - W_{10}^{k}} = e^{-j\left(\frac{4\pi k}{10}\right)} \frac{\sin(\pi k/2)}{\sin(\pi k/10)}$$

$$\frac{1 - e^{-j\left(\frac{2\pi}{10}\right)5k}}{1 - e^{-j\left(\frac{2\pi}{10}\right)k}} = \frac{1 - e^{-j\pi k}}{1 - e^{-j\frac{\pi}{5}k}} = \frac{e^{-j\frac{\pi}{2}k}\left(e^{j\frac{\pi}{2}k} - e^{-j\frac{\pi}{2}k}\right)}{e^{-j\frac{\pi}{10}k}\left(e^{j\frac{\pi}{10}k} - e^{-j\frac{\pi}{10}k}\right)} = e^{-j\left(\frac{4\pi k}{10}\right)}\left\{\frac{e^{j\frac{\pi}{2}k} - e^{-j\frac{\pi}{2}k}}{e^{j\pi/10\,k} - e^{-j\frac{\pi}{10}k}}\right\}$$

$$\tilde{X}(k) = \underbrace{e^{-j\frac{4\pi k}{10}}}_{\text{phase}} \; \underbrace{\frac{\sin\frac{\pi}{2}k}{\sin\frac{\pi}{10}k}}_{\text{Magnitude}}$$



$\times \rightarrow$ Inderminate (magnitude $= 0$).

key result $\Rightarrow$ analysis - synthesis DFS pair $\Rightarrow$ basis for DFT.

Discrete Fourier Transform

$x(n) = 0 \;,\; n < 0 \;,\; n > N-1$

Finite length N

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n + rN)$$

$$= x(n \text{ modulo } N)$$

$$\triangleq x((n))_N$$

e.g for $N = 7$

$((25))_7 = 4$

$((-(6))_7 = 5$

In Matlab $\rightarrow$ mod( ).

$x(n) = \tilde{x}(n) \, R_N(n)$

$\tilde{X}(k) = $ DFS coeff. of $\tilde{x}(n)$.

DFS

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) \, W_N^{nk}$$

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) \, W_N^{-kn}$$

DFT.

$$\text{DFT} \leftarrow X(k) = \begin{cases} \sum_{n=0}^{N-1} x(n) W_N^{nk} & , k=0,1,\dots,N-1 \\ 0 & , \text{otherwise} \end{cases}$$

finite

$$X(k) = \tilde{X}(k) R_N(k)$$

$\tilde{X}(k) = X((k))_N$ periodically repeated $X(k)$

$$X(k) = \left[\sum_{n=0}^{N-1} x(n) W_N^{nk}\right] R_N(k)$$

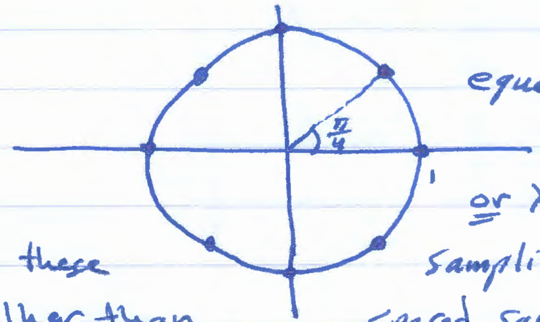
$$x(n) = \left[\frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-nk}\right] R_N(n)$$

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n}$$

$\rightarrow$ because $x(n)=0$ outside $[0, N-1]$

so, $$X(k) = X(z)\Big|_{z=W_N^{-k}} \quad , k=0,1,2,\dots,N-1$$

$*$ $X(k)$ samples of Z-transform on unit-circle because $|W_N^{-k}| = 1$



equally spaced in angle.

e.g $N=8$

or $X(k)$ (DFT) is sampling $X(e^{jw})$ N equally spaced samples by $\frac{2\pi}{N}$ $(0 \rightarrow 2\pi)$

multiplying by

$*$ $R_N(n) \longrightarrow$ means that we take these samples only one round. rather than running around unit circle again and again.

Properities of DFT

* Focus on difference between properities of DFT and (FT and Z-transform)

* <u>Shifting</u>.

$$x(n) \xleftrightarrow{DFT} X(k)$$

$$\tilde{x}(n) \xleftrightarrow{DFS} \tilde{X}(k) \rightarrow \text{periodic extension of } X(k) \text{ (DFT).}$$

$$\tilde{x}_1(n) = \tilde{x}(n+m) \xleftrightarrow{DFS} \tilde{X}(k)\, W_N^{-km}$$

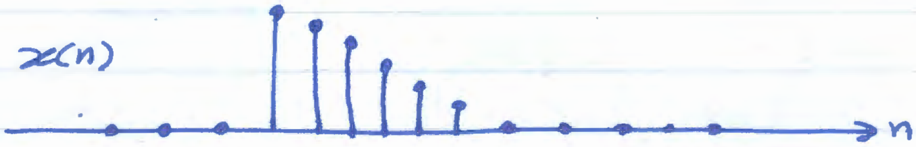
$$x_1(n) \xleftrightarrow{DFT} X(k)\, W_N^{-km}$$

extract one period of shifted periodic seq. $\tilde{x}_1(n)$.

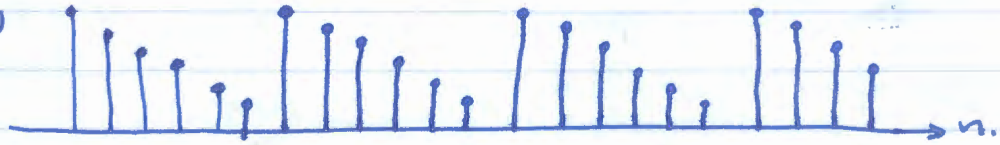
* Shifting finite-length seq. is different than shifting periodic seq. because finite-length seq. is zero outside of Interval $[0, N-1]$ so, by shifting, $x(n)$ can be no longer zero's outside $[0, N-1]$,

⇒ Therefore we use circular shifting for $x(n)$ and not linear shifting.

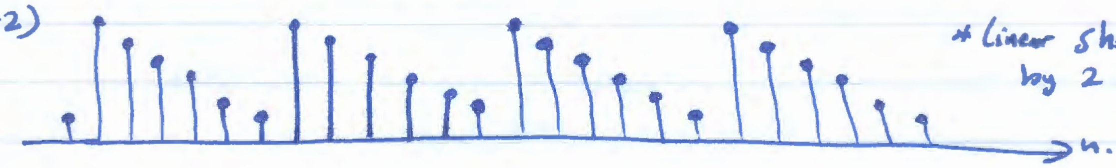
$x(n)$



$\tilde{x}(n)$

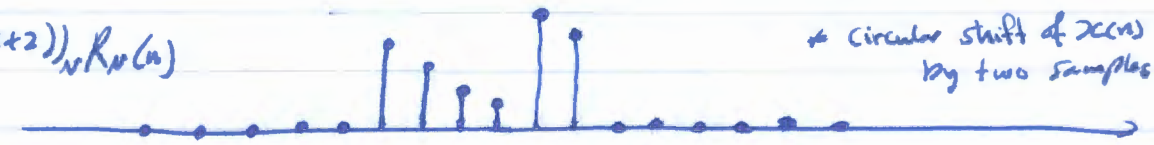


$\tilde{x}(n+2)$



* linear shift of $\tilde{x}(n)$ by 2 samples.

$x((n+2))_N R_N(n)$



* circular shift of $x(n)$ in $[0, N-1]$ by two samples.

Duality

$$x((n+m))_N R_N(n) \xleftrightarrow{DFT} W_N^{-km} X(k)$$

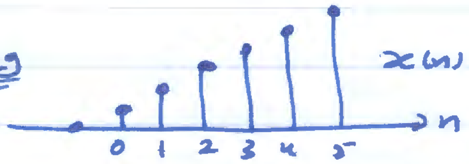
$$W_N^{Ln} x(n) \xleftrightarrow{DFT} X((k+L))_N R_N(k)$$

Cont.

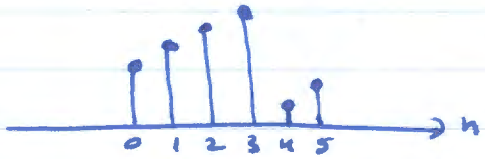
* DFT ~~Symmetry~~ properties.

* Circular shift : $x((n-m))_N \,,\; \underbrace{0 \le n \le N-1}_{R_N(n)} \longleftrightarrow W_N^{-km} X(k)$

e.g

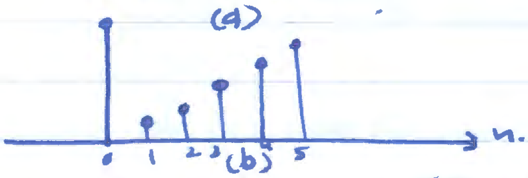


(a)



$x((n-4))_6 = x((n+2))_6$

(c)

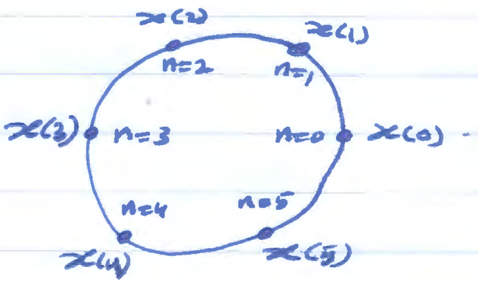


(b)

$x((n-1))_6 = x((n+5))_6$

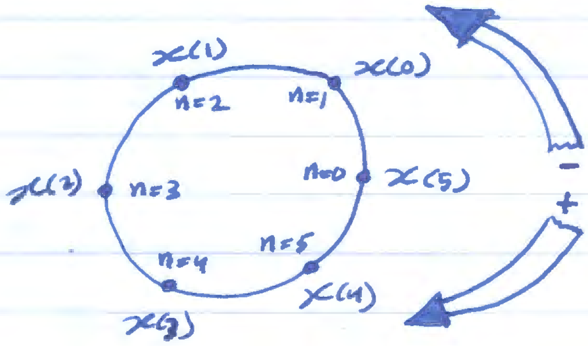
$$x((-n))_N = x((N-n))_N$$

$$x((-n))_N = \begin{cases} x(N-n), & \text{for } 1 \le n \le N-1 \\ x(n), & \text{for } n=0 \end{cases}$$

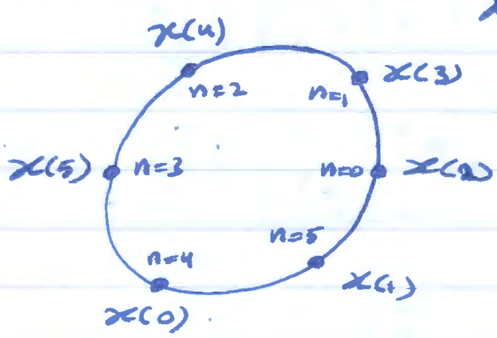


(a)

x(n)



$x((n-1))_6 = x((n+5))_6$



$x((n-4))_6 = x((n+2))_6$ ..

* Symmetry Property

DFS $\tilde{x}(n)$ real

$$\tilde{X}_R(k) = \tilde{X}_R(N-k)$$
$$\tilde{X}_I(k) = -\tilde{X}_I(N-k)$$

DFT $x(n)$ real

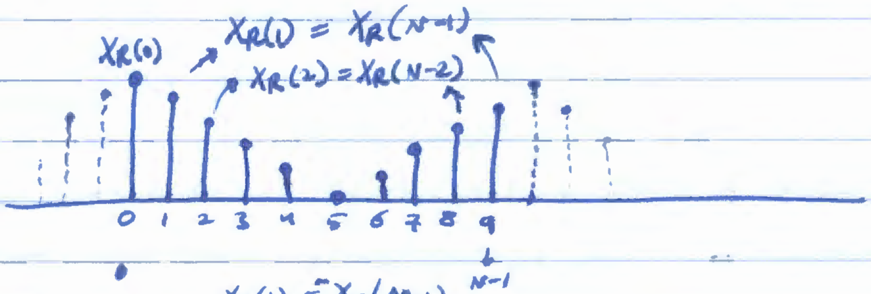
$$X_R(k) = X_R((N-k))_N \cdot R_N(n) \quad \text{(even)}$$

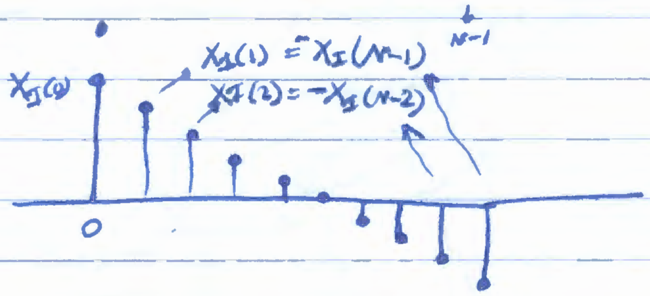
$$X_I(k) = -X_I((N-k))_N \cdot R_N(n) \quad \text{(odd)}$$

Example:

$$X_R(0) = X_R((N-0))_N \cdot \cancel{R_N(0)}^{1} = X_R(0)$$

$$X_R(1) = X_R((N-1))_N \cdot \cancel{R_N(1)}^{1} = X_R(N-1)$$





* Duality

If N-point DFT of length-N seq. $x(n)$ is $X(k)$, then the N-point DFT of length-N seq. $X(n)$ is given by $N\,x((-k))_N$

$$X(n) \xleftrightarrow{DFT} N\,x((-k))_N, \quad 0 \le k \le N-1$$

* Convolution property (Circular convolution).

$$x_3(n) \xleftrightarrow{DFT} X_1(k) \cdot X_2(k)$$

we know:

$$\tilde{x}(n) \xleftrightarrow{DFS} \tilde{X}_1(k)\, \tilde{X}_2(k)$$

$$x_3(n) = \tilde{x}_3(n)\, R_N(n)$$

$$x_3(n) = \left[\sum_{m=0}^{N-1} \tilde{x}_1(m)\, \tilde{x}_2(n-m) \right] R_N(n)$$

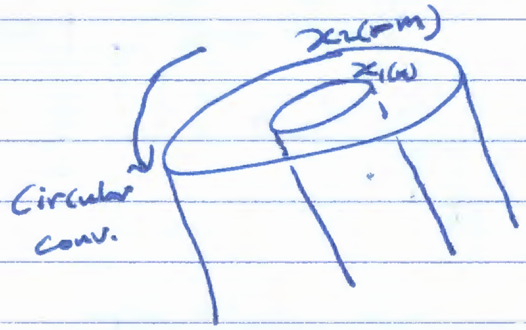
$$= \left\{ \sum_{m=0}^{N-1} \underbrace{x_1((m))_N}_{x_1(m)} \; \underbrace{x_2((n-m))_N}_{\text{Flip + circular shift.}} \right\} R_N(n)$$

$$x_3(n) = x_1(n) \; \text{(N)} \; x_2(n)$$

↳ N-point circular convolution

$$x(n) \rightarrow \boxed{h(n)} \rightarrow y(n)$$

↳ can be implemented by computing DFT of h(n) and input x(n) then multiply them and then take Inverse DFT ⇒ corresponds to circular convolution.



Rotate multiply and ADD.

* linear convolution

we flip one seq. and we shift it toward the other, then multiply and ADD

$$x_3(n) = x_1(n) \; \text{Ⓝ} \; x_2(n)$$

$$= \left[\sum_{m=0}^{N-1} x_1(m)\, \tilde{x}_2(n-m) \right] R_N(n)$$

$$= \left\{ \sum_{m=0}^{N-1} x_1(m)\, x_2((n-m))_N \right\} R_N(n)$$

$$y(n) = x(n) \; \text{Ⓝ} \; h(n) = h(n) \; \text{Ⓝ} \; x(n)$$

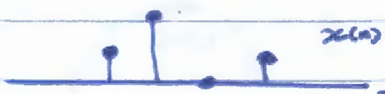
* N-point circular convolution can be written in matrix form:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \underbrace{\begin{bmatrix} h(0) & h(N-1) & h(N-2) & \cdots & h(1) \\ h(1) & h(0) & h(N-1) & \cdots & h(2) \\ h(2) & h(1) & h(0) & \cdots & h(3) \\ \vdots & \vdots & \vdots & & \vdots \\ h(N-1) & h(N-2) & h(N-3) & \cdots & h(0) \end{bmatrix}}_{\text{Circulant Matrix}} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{bmatrix}$$

* elements in each row are obtained by circularly rotating the elements of previous row to the right by 1.

Ex Determine 4-point circular convolution of two length-4 sequences $x(n)$ and $h(n)$.

$$x(n) = \{1 \;\; 2 \;\; 0 \;\; 1\}, \quad h(n) = \{2 \;\; 2 \;\; 1 \;\; 1\}, \quad 0 \le n \le 3.$$



$$y(n) = x(n) \; ④ \; h(n) = \sum_{m=0}^{3} x(m)\, h((n-m))_4, \quad 0 \le n \le 3$$

$$y(0) = \sum_{m=0}^{3} x(m)\, h((-m))_4$$

The circularly time reversed seq. $h((-m))_4$ is

$$h((-m))_4 = \{h(0)\ h(3)\ h(2)\ h(1)\} = \{2\ 1\ 1\ 2\}$$

$$\sum_{m=0}^{3} x(m)\, h((-m))_4 = x(0)h(0) + x(1)h(3) + x(2)h(2) + x(3)h(1) = 6$$

Similarly,

$$y_c(1) = \sum_{m=0}^{3} x(m)\, h((1-m))_4$$

* $h((1-m))_4$ is obtained by circularly shift $h((-m))$ to right by one sample.

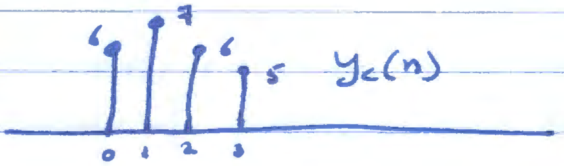
$$h((1-m))_4 = \{h(1)\ h(0)\ h(3)\ h(2)\} = \{2\ 2\ 1\ 1\}$$

So,

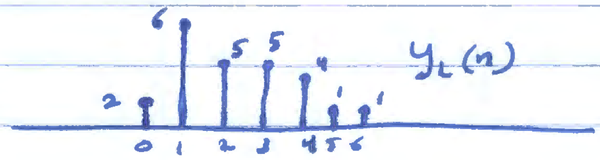
$$y_c(1) = x(0)h(1) + x(1)h(0) + x(2)h(3) + x(3)h(2) = 7$$

$$y_c(2) = x(0)h(2) + x(1)h(1) + x(2)h(0) + x(3)h(3) = 6$$

$$y_c(3) = x(0)h(3) + x(1)h(2) + x(2)h(1) + x(3)h(0) = 5$$

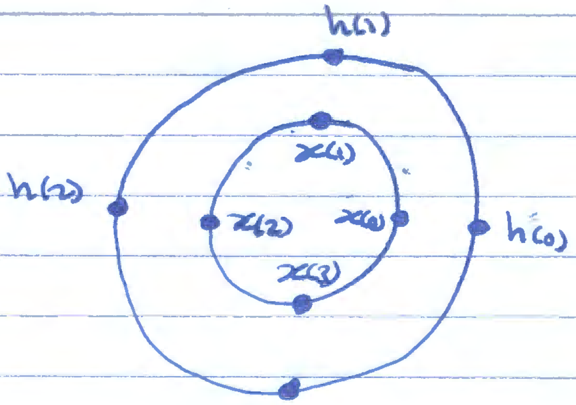


Circular conv.

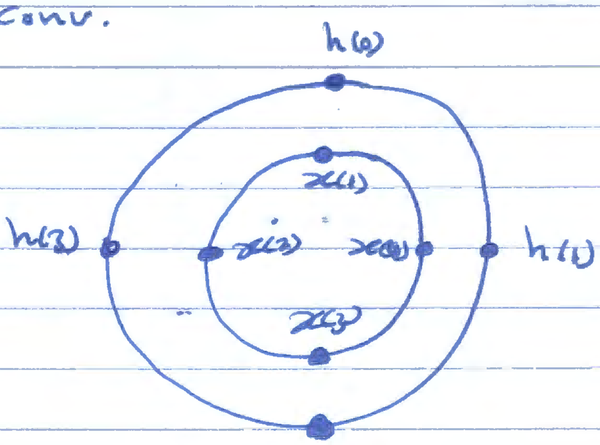


Linear conv.

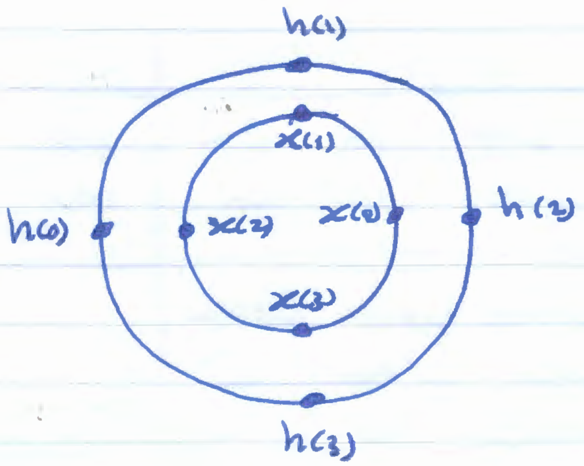
* graphical representation of Circular conv.



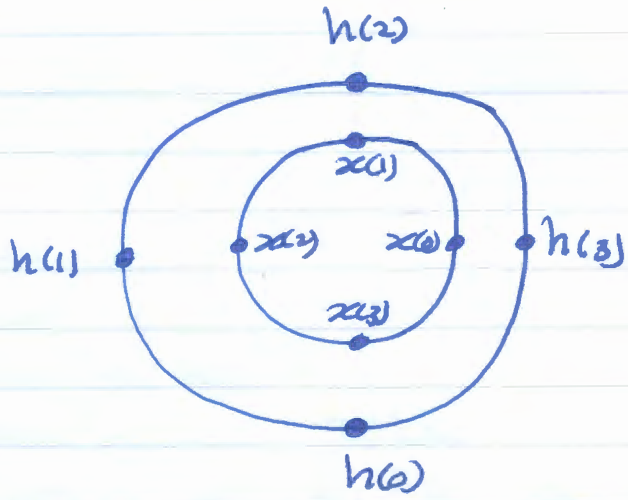
$y_c(0)$



$y_c(1)$



$y_c(2)$



$y_c(3)$.

* Circularly time-reversed and shifted seq. $h(n-m)$.

* Circular Convolution using Matlab:

```
g=[1 2 0 1]; h=[2 2 1 1];
y=circonv(g,h); % y=[6 7 6 5]
```

* Tabular Method for computing Circular Conv.

$y(n) = g(n) \, N \, h(n)$

Ex.

$g(n) = \{1 \;\; 2 \;\; 0 \;\; 1\}, \quad h(n) = \{2 \;\; 2 \;\; 1 \;\; 1\}$.

Find $y(n) = g(n) \, 4 \, h(n)$.

n	0	1	2	3	$((4))_4$	$((5))_4$	$((6))_4$
g(n)	g(0)	g(1)	g(2)	g(3)			
h(n)	h(0)	h(1)	h(2)	h(3)			
Circular shift 4-times to left	g(0) h(0)	g(1) h(0)	g(2) h(0)	g(3) h(0)			
	–	g(0) h(1)	g(1) h(1)	g(2) h(1)	g(3) h(1)		
	–	–	g(0) h(2)	g(1) h(2)	g(2) h(2)	g(3) h(2)	
	–	–	–	g(0) h(3)	g(1) h(3)	g(2) h(3)	g(3) h(3)

n:	0	1	2	3
g(n):	g(0)	g(1)	g(2)	g(3)
h(n):	h(0)	h(1)	h(2)	h(3)
$\sum$	g(0) h(0)	g(1) h(0)	g(2) h(0)	g(3) h(0)
	g(3) h(1)	g(0) h(1)	g(1) h(1)	g(2) h(1)
	g(2) h(2)	g(3) h(2)	g(0) h(2)	g(1) h(2)
	g(1) h(3)	g(2) h(3)	g(3) h(3)	g(0) h(3)
$y_c(n)$:	$y_c(0)$	$y_c(1)$	$y_c(2)$	$y_c(3)$

n:	0	1	2	3
g(n):	1	2	0	1
h(n):	2	2	1	1
$\sum$	2	4	0	2
	2	2	4	0
	0	2	1	2
	2	0	1	1
$y_c(n) =$	6	7	6	5

* <u>Circular Convolution property</u>

$$x(n) \, Ⓝ \, h(n) \xleftrightarrow{DFT} X(k)H(k)$$

<u>proof</u>

let $y(n) = x(n) \, Ⓝ \, h(n)$

$$Y(k) = \sum_{n=0}^{N-1} y(n) W_N^{kn} = \sum_{n=0}^{N-1} \left[\sum_{m=0}^{N-1} x(m) h((n-m))_N \right] W_N^{kn}$$

*by interchanging order of Sum's and substitube $\boxed{n-m = L + Nr}$ which lead to r integer with $0 \leq L \leq N-1$

$$Y(k) = \sum_{m=0}^{N-1} x(m) \left[\sum_{n=0}^{N-1} h((n-m))_N W_N^{kn} \right]$$

$$= \sum_{m=0}^{N-1} x(m) \left[\sum_{L=0}^{N-1} h(L) W_N^{k(L+m+Nr)} \right]$$

$$= \sum_{m=0}^{N-1} x(m) \left[\sum_{L=0}^{N-1} h(L) W_N^{kL} \right] W_N^{km}$$

$$= \left\{ \sum_{m=0}^{N-1} x(m) W_N^{km} \right\} \left\{ H(k) \right\} = X(k) H(\cancel{k}\,k)$$

<u>Note 1</u>

If we let $r = ((m))_N$

$r = m + LN$, L is Integer choosen to make $m+LN$ a number between 0 and $N-1$

~~Relationsh~~

Relationship between Circular Conv. and Linear Conv.

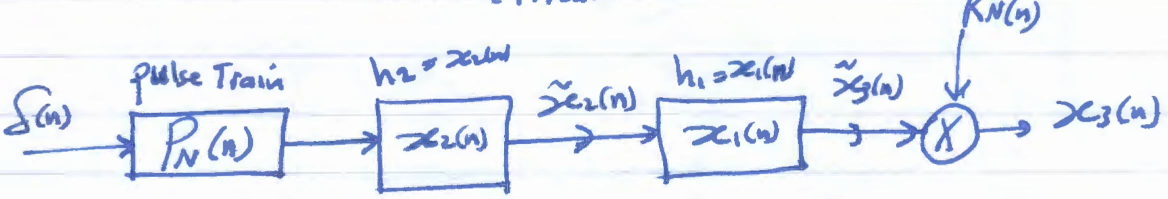
$$x_3(n) = x_1(n) \; (N) \; x_2(n)$$

$$= \left[\sum_{m=0}^{N-1} x_1(m) \, x_2((n-m))_N \right] R_N(n)$$

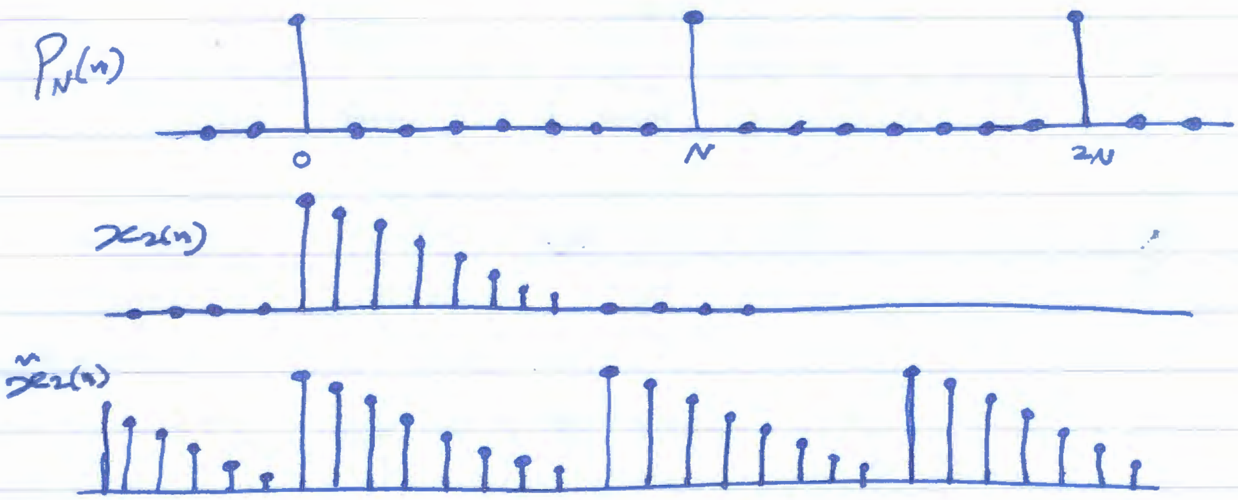
↗ periodic seq. of $x_2(n)$

$$x_3(n) = \left[x_1(n) * x_2((n))_N \right] R_N(n)$$

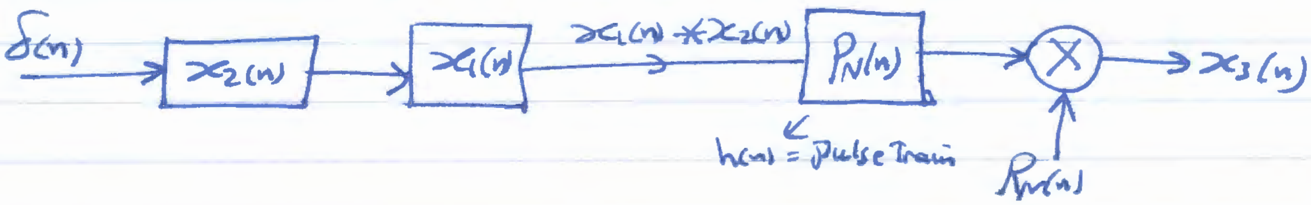
↓ Linear Conv.



Cascade of 3 LTI sys. (Circular Conv. of x_1 and x_2)



* LTI systems in Cascade ⇒ so we can change order



* This means: that we can form circular conv. between x_1 and x_2 by linear convolution of x_1 and x_2 and then convolve result with impulse train, and then extract one period.

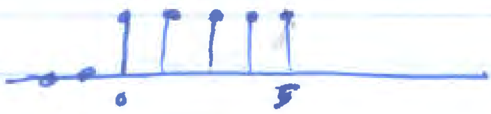
Or Its a linear conv. $(x_1 * x_2)$ + aliasing. (repeat it over and over again)

$$\hat{x}_3(n) = x_1(n) * x_2(n)$$

$$x_3(n) = x_1(n) \, (N) \, x_2(n)$$

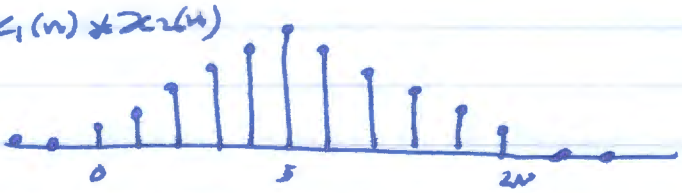
$$x_3(n) = \left\{ \sum_{r=-\infty}^{\infty} \hat{x}_3(n+rN) \right\} R_N(n)$$

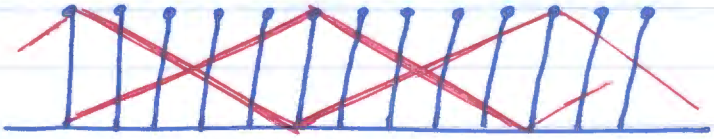
Ex. $x_1(n) = x_2(n)$



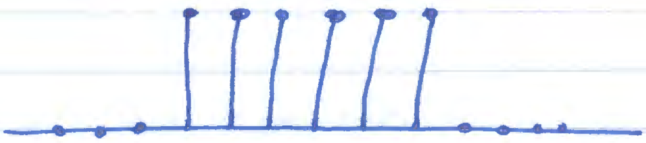
Do $x_1(n) \, (N) \, x_2(n)$ by doing $x_1(n) * x_2(n)$ + aliasing.

$x_1(n) * x_2(n)$





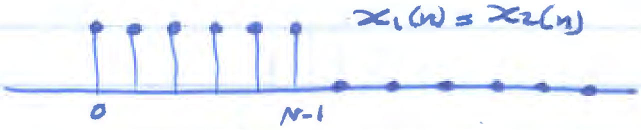
$x_1(n) * x_2(n) * P_N(n)$



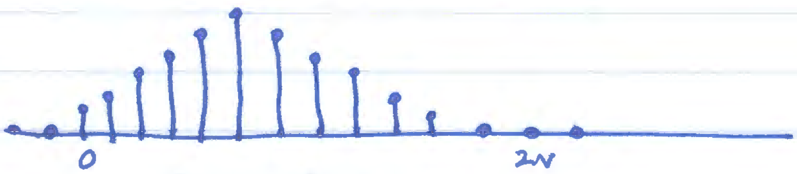
$x_1(n) \, (N) \, x_2(n)$

* Taking 2N-point

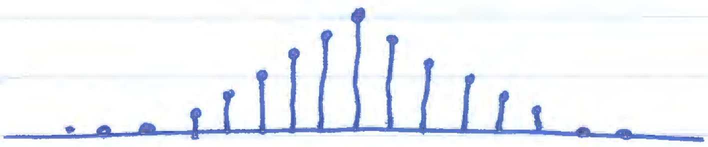
$x_1(n) = x_2(n)$



$x_1(n) * x_2(n) * P_{2N}(n)$



$x_1(n) * x_2(n)$



$x_1(n) \, (2N) \, x_2(n)$

* we can implement linear convolution by circular convolution if we take length of circular conv. long enough

$x_1(n) \rightarrow$ length $= N$, $x_2(n) \rightarrow$ length $= M \Rightarrow$ circular conv. on $\lfloor N+M-1 \rfloor \Rightarrow$ This refered to padding with zeros.

$\longrightarrow$ For very long sequences:

- overlap-Add Method (sectioning) $\rightarrow$ linear conv.
- overlap-save Method (circular conv.).

overlap-ADD:

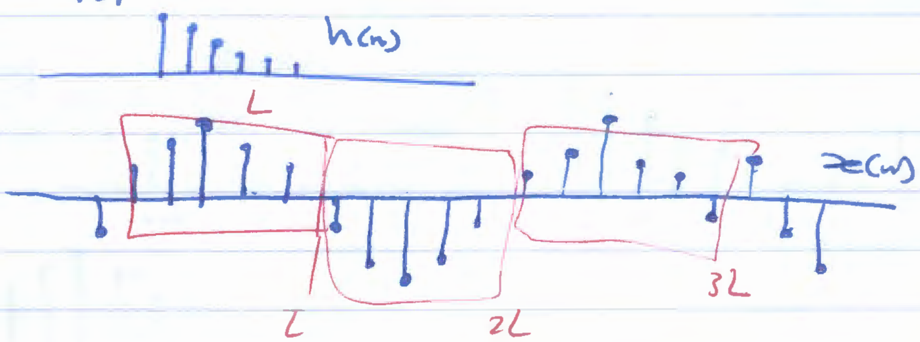
sectioning long seq. into short segments

$$x(n) = \sum_{i=1}^{S} x_i(n)$$ i.e. $x_0(n), x_1(n), x_2(n), \dots$ all with length L.

$\Rightarrow$ we know convolution of sum = sum of convolutions.

suppose $h(n)$ is of length $-M$
linear conv. $h(n) * x_i(n) \rightarrow$ length $L+M+1$
Then, we add result together.

$$\sum_{i=1}^{S} h(n) * x_i(n)$$ $\rightarrow$ There will be overlap.



(see this example on slide).

DFT Computation:

DFT: $X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}$, $0 \le k \le N-1$

IDFT (Inverse DFT):

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-kn} \quad , 0 \le n \le N-1$$

Example:

$$x(n) = \begin{cases} 1, & n=0 \\ 0, & 1 \le n \le N-1 \end{cases}$$

N-point DFT of $x(n)$ is $X(k) = 1$.

Ex:

$$y(n) = \begin{cases} 1, & n=m, \quad 0 \le m \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

$$\Rightarrow Y(k) = W_N^{km}$$

* Matrix Relation

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad , \quad 0 \le k \le N-1$$

$$\mathbf{X} = D_N x$$

$\mathbf{X} \rightarrow$ vector composed of N DFT samples.

$$X = [X(0) \; X(1) \; X(2) \; \dots \; X(N-1)]^T$$

$x \rightarrow$ vector of N input samples.

$$x = [x(0) \; x(1) \; x(2) \; \dots \; x(N-1)]^T$$

$D_N \rightarrow N \times N$ DFT Matrix

$$D_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & & & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

Likewise IDFT can be expressed in Matrix form:

$$\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} = D_N^{-1} \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}$$

D_N^{-1} is NXN IDFT Matrix.

$$D_N^{-1} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\ 1 & W_N^{-2} & W_N^{-4} & \cdots & W_N^{-2(N-1)} \\ \vdots & \vdots & & & \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \cdots & W_N^{-(N-1)(N-1)} \end{bmatrix}$$

$$D_N^{-1} = \frac{1}{N} D_N^{*}$$

*DFT computation in Matlab.

- fft(x) , fft(x, N), ifft(X), ifft(X, M)
- DFTmtx(N) → compute NXN DFT Matrix (D_N).
- To compute D_N^{-1} ⇒ conj(dftmtx(N))/N.

<u>Example</u>: $g(n) = \{1 \ 2 \ 0 \ 1\}$

$h(n) = \{2 \ 2 \ 1 \ 1\}, \ 0 \le n \le 3$

$$G(k) = g(0) + g(1)\, e^{-j\frac{2\pi k}{4}} + g(2)\, e^{j\frac{4\pi k}{4}} + g(3)\, e^{-j\frac{6\pi k}{4}}$$

$$= 1 + 2e^{-j\frac{\pi k}{2}} + e^{-j\frac{3\pi k}{3}}, \quad k = 0, 1, 2, 3$$

Therefore,

$$G(0) = 1+2+1 = 4$$
$$G(1) = 1-j2+j = 1-j$$
$$G(2) = 1-2-1 = -2$$
$$G(3) = 1+j2-j = 1+j$$

The above DFT can also be computed using matrix relation.

$$\begin{bmatrix} G(0) \\ G(1) \\ G(2) \\ G(3) \end{bmatrix} = D_4 \begin{bmatrix} g(0) \\ g(1) \\ g(2) \\ g(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 1-j \\ -2 \\ 1+j \end{bmatrix}$$

likewise DFT of $h(n)$ is:

$$\begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 1-j \\ 0 \\ 1+j \end{bmatrix}$$

* Applying 4-point IDFT to the Product $G(k)H(k) = Y_c(k)$

$$\begin{bmatrix} Y_c(0) \\ Y_c(1) \\ Y_c(2) \\ Y_c(3) \end{bmatrix} = \begin{bmatrix} G(0)H(0) \\ G(1)H(1) \\ G(2)H(2) \\ G(3)H(3) \end{bmatrix} = \begin{bmatrix} 24 \\ -j2 \\ 0 \\ j2 \end{bmatrix}$$

* we arrive at the desired circular convolution result:

$$\begin{bmatrix} y_c(0) \\ y_c(1) \\ y_c(2) \\ y_c(3) \end{bmatrix} = \frac{1}{4} D_4^* \begin{bmatrix} 24 \\ -j2 \\ 0 \\ j2 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 24 \\ -j2 \\ 0 \\ j2 \end{bmatrix} = \begin{bmatrix} 6 \\ 7 \\ 6 \\ 5 \end{bmatrix}$$